# 1

# Introduction

It is an expensive process to rank and select the best one from many complex discrete event dynamic systems that are computationally intensive to simulate. Therefore, learning the optimal system, action, alternative, candidate, or design is a classical problem and has many applications in the areas of intelligent system design, statistics and stochastic simulation, machine learning, and artificial intelligence.

Stochastic ranking and selection of given system designs can be treated as a simulation optimization problem. Its solution requires one to design statistical procedures that can select the one with the highest mean performance from a finite set of systems, alternatives, candidates, or designs. Their mean values are unknown and can be estimated only by statistical sampling, because their reward from environments is not deterministic but stochastic due to unknown noise. It is a classical problem in the areas of statistics and stochastic simulation.

**Example 1.1** Finding the most effective drug or treatment from many different alternatives where the economic cost of each sample for testing the effectiveness of the drug is very expensive and risky. Another representative example is the archery competition. How can we select the real champion while using the least number of arrows?

Noise comes mainly from three kinds of uncertainties [1]. (i) Environmental uncertainties: operating temperature, pressure, humidity, changing material properties and drift, etc. are a common kind of uncertainties. (ii) Design parameter uncertainties: the design parameters of a product can only be realized to a certain degree of accuracy because high precision machinery is expensive. (iii) Evaluation uncertainties: the uncertainty happens in the evaluation of the system output and the system performance including measuring errors and all kinds of approximation errors if models instead of the real physical objects are used.

*Learning Automata and Their Applications to Intelligent Systems*, First Edition.
JunQi Zhang and MengChu Zhou.

## 1.1 Ranking and Selection in Noisy Optimization

Real-world optimization problems are often subject to uncertainty as variables can be affected by imprecise measurements or just corrupted by other factors such as communication errors. In either case, uncertainty is an inherent characteristic of many such problems and therefore needs to be considered when tailoring metaheuristics to find good solutions. Noise is a class of uncertainty and corrupts the objective values of solutions at each evaluation [2]. Noise has been shown to significantly deteriorate the performance of different metaheuristics such as Particle Swarm Optimizer (PSO), Genetic Algorithms (GA), Evolutionary Strategies (ES), Differential Evolution (DE), and other metaheuristics.

**Example 1.2** Ranking and selection are the critical part in PSO. First, each particle has to compare the fitness of its new position to its previous best and retain the better one. Second, the overall best solution found so far has to be determined as a global best solution to lead the swarm flying and refining the accuracy. Since PSO has a memory that stores the estimated personal best solution of a particle and the estimated global best solution of a swarm, noise leads such memory to be inaccurate over iterations and particles eventually fail to rank and select good solutions from bad ones, which drives the particle swarm toward a wrong direction. Concretely, a noisy fitness function induces noisy fitness evaluations and causes two types of undesirable selection behavior [3]: (i) A superior candidate may be erroneously believed to be inferior, causing it to be eliminated; (ii) An inferior candidate may be erroneously believed to be superior, causing it to survive and reproduce. These behaviors in turn cause the following undesirable effects: (i) The learning rate is reduced. (ii) The system does not retain what it has learnt. (iii) Exploitation is limited. (iv) Fitness does not monotonically improve with generation even with elitism.

Uncertainty has to be taken into account in many real-world optimization problems. For example, in the engineering field, the signal returned from the real world usually includes a significant amount of noise due to the measure error or various uncertainties. It is usually modeled as sampling noise from a Gaussian distribution [4]. Therefore, the noise can be characterized by its standard deviation $\sigma$ and classified into additive and multiplicative ones. Its impact to function $\check{F}(x)$ can be expressed as:

$$\hat{F}_{+}(\boldsymbol{x}) = \check{F}(\boldsymbol{x}) + N(0, \sigma^2), \tag{1.1}$$

$$\hat{F}_{\times}(\boldsymbol{x}) = \check{F}(\boldsymbol{x}) \times N(1, \sigma^2), \tag{1.2}$$

where $\check{F}(x)$ represents the real fitness value of solution $x$, $\hat{F}_{+}(x)$ and $\hat{F}_{\times}(x)$ are illusory fitness values of solution $x$ in additive and multiplicative noisy environments,

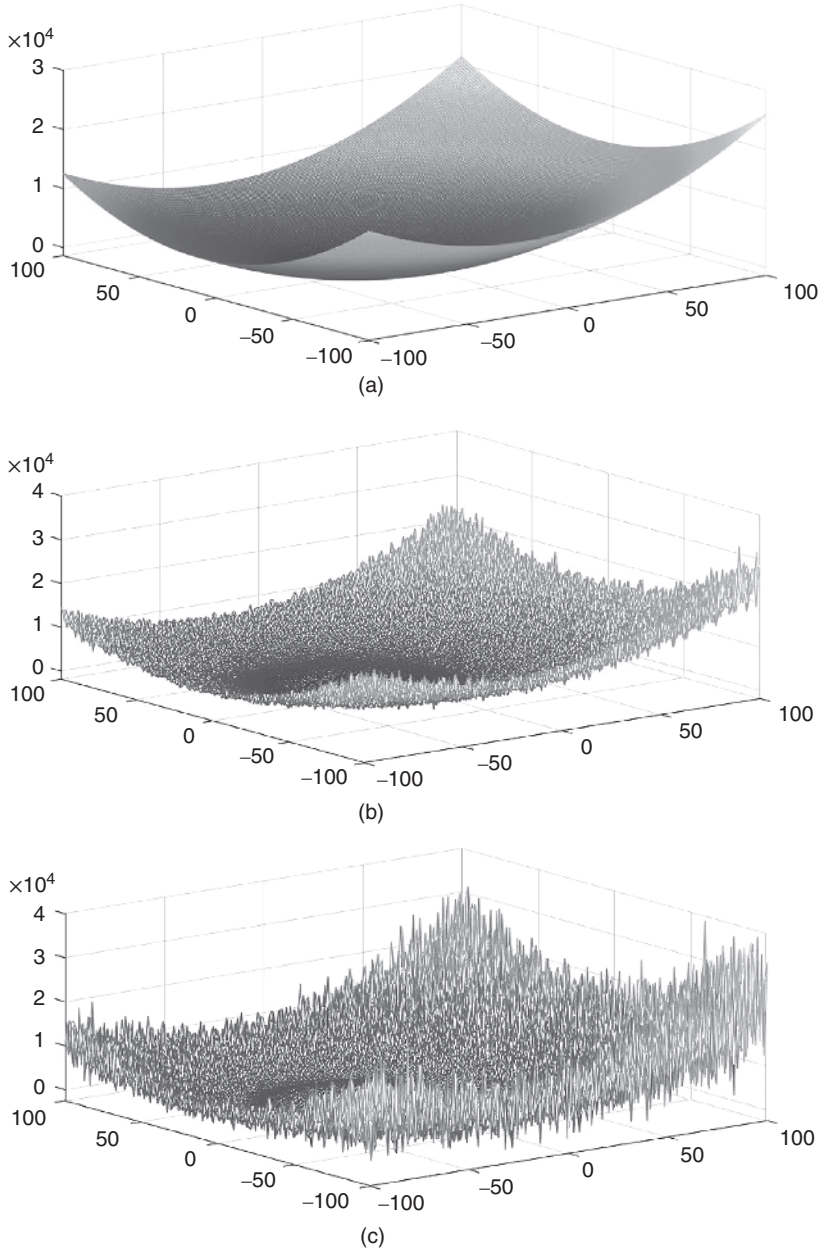


**Figure 1.1** 3-D map of a Sphere function with additive and multiplicative noise. (a) True. (b) Corrupted by additive noise with $\sigma = 0.1$. (c) Corrupted by multiplicative noise with $\sigma = 0.3$.

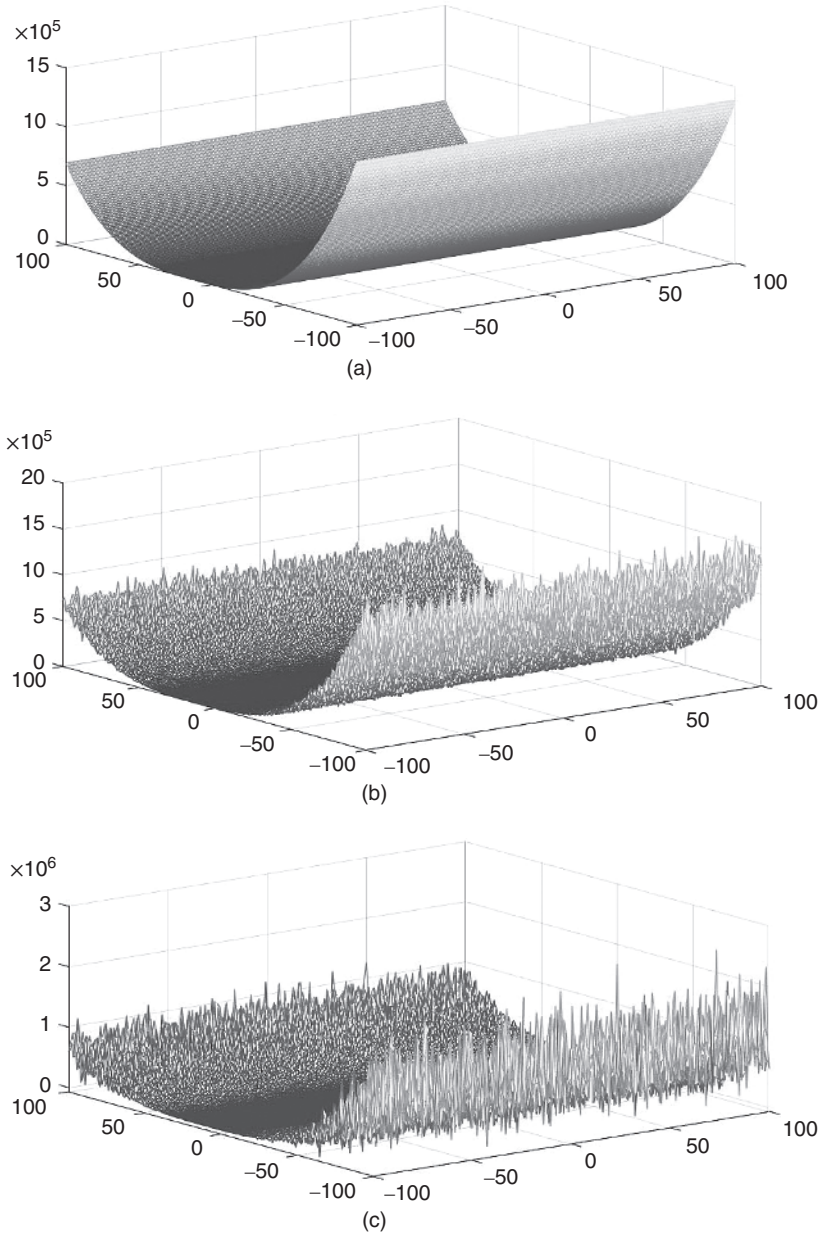


**Figure 1.2** 3-D map of a Different Powers function with additive and multiplicative noise. (a) True. (b) Corrupted by additive noise with $\sigma = 0.1$. (c) Corrupted by multiplicative noise with $\sigma = 0.3$.

respectively. It is worth noting that they are identical in the environment where $\sigma = 0$. It is obvious that multiplicative noise is much more challenging, since it can bring much larger disturbance to fitness values than additive noise.

**Example 1.3** The 3-D maps of functions [5] with additive and multiplicative noise with different $\sigma$ are illustrated in Figs. 1.1 and 1.2. The figures show that the additional challenge of multiplicative noise over additive noise is a larger corruption of the objective values whose magnitude changes across the search space proportionally to the objective values of the solutions. For multiplicative noise, its impact depends on the optimization objective and the range of objective space. Specifically, on minimization problems whose objective space is only positive, the objective values of better solutions (whose fitness value is small) can be less affected by multiplicative noise. Conversely, in maximization problems, the objective values of better solutions can be more affected [2].

Therefore, if the problem is subject to noise, the quality of the solutions deteriorates significantly. The basic resampling method uses many re-evaluations to estimate the fitness of a candidate solution. Thus, the efficiency of resampling determines the accuracy of ranking and selection of the elite solutions, which are critical to intelligent optimization methods during their evolution to the optimal solutions. Yet the resampling costs too much. Considering a fixed and limited computational budget of function evaluations or environmental interactions, resampling methods better estimate the objective values of the solutions by performing fewer iterations. Intelligent allocation of resampling budget to candidate solutions can save many evaluations and improve the estimation of the solution fitness.

## 1.2 Learning Automata and Ordinal Optimization

Ranking and selection procedures were developed in the 1950s for statistical selection problems such as choosing the best treatment for a medical condition [6]. The problem of selecting the best among a finite set of alternatives needs an efficient learning scheme given a noisy environment and limited simulation budget, where the best is defined with respect to the highest mean performance, and where the performance is uncertain but may be estimated via simulation. Approaches presented in the literature include frequentist statistics, Bayesian statistics, heuristics [7], and asymptotic convergence in probability [6]. This book focuses on learning automata and ordinal optimization methods to solve stochastic ranking and selection problems.

Investigation of a Learning Automaton (LA) began in the erstwhile Soviet Union with the work of Tsetlin [8, 9] and popularized as LA in a survey paper

in 1974 [10]. These early models were referred to as deterministic and stochastic automata operating in random environments. Systems built with LA have been successfully employed in many difficult learning situations and the reinforcement learning represents a development closely related to the work on LA over the years. This has also led to the concept of LA being generalized in a number of directions in order to handle various learning problems.

An LA represents an important tool in the area of reinforcement learning and aims at learning the optimal action that maximizes the probability of being rewarded out of a set of allowable actions by the interaction with a random environment. During a cycle, an automaton chooses an action and then receives a stochastic response that can be either a reward or penalty from the environment. The action probability vector of choosing the next action is then updated by employing this response. The ability of learning how to choose the optimal action endows LA with high adaptability to the environment. Various LAs and their applications have been reviewed in survey papers [10], [11] and books [12–15].

Ordinal Optimization (OO) is introduced by Ho et al. [16]. There are two basic ideas behind it: (i) Estimating the order among solutions is much easier than estimating the absolute objective values of each solution. (ii) Softening the optimization goal and accepting good enough solutions leads to an exponential reduction in computational burden. The Optimal Computing Budget Allocation (OCBA) [17–22] is a famous approach that uses an average-case analysis rather than a worst-case bound of the indifference zone. It attempts to sequentially maximize the probability that the best alternative can be correctly identified after the next stage of sampling. Its procedure tends to require much less sampling effort to achieve the same or better empirical performance for correct selection than the procedures that are statistically more conservative.

**Example 1.4** This example shows the objective of ordinal optimization. A system works with one of 10 components that have their own time-to-failure. The objective is to decide which component is the worst one to minimize steady-state system unavailability given budget constraints. We have the budget to perform only 50 component tests to obtain if the tested component leads to a failure. How do we allocate these budget to these 10 components so as to identify the worst one is the objective of ordinal optimization.

The next example shows the reason of softening the optimization goal, i.e., reducing computational burden.

**Example 1.5** This example shows the reason of softening the optimization goal and accepting good enough solutions leads to an exponential reduction in computational burden. A class consists of 50 students. The objective is to find the tallest one. However, we do not have to measure all student's accurate height. We can use some sorting algorithms like bubble sorting to identify the tallest student. In this

way, we do not need all students' accurate heights. Furthermore, the order of the other students except the tallest one is not important and can be inaccurate. An ordinal optimization method can also be used to identify the tallest student.

## 1.3 Exercises

1 What is the objective of stochastic ranking and selection?

2 What are the difficulties caused by noise in solving optimization problems?

3 What are the advantages and constraints of resampling in noisy optimization?

4 What is the objective of a learning automaton?

5 What are the basic ideas of ordinal optimization?

6 For additive and multiplicative noise, which one is more difficult to handle and why?

7 In Example 1.5., how to find the tallest student without their accurate height data?

## References

1 H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.
2 J. Rada-Vilela, "Population statistics for particle swarm optimization on problems subject to noise," Ph.D. dissertation, 2014.
3 A. Di Pietro, L. While, and L. Barone, "Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions," in *Proceedings of Congress on Evolutionary Computation*, vol. 2. IEEE, 2004, pp. 1254–1261.
4 Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
5 J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, vol. 201212, 2013.
6 M. C. Fu, "Handbook of simulation optimization," Springer, 2015, vol. 216.

7 Y. Jin, H. Wang, and C. Su, "Data-driven evolutionary optimization integrating evolutionary computation," Machine Learning and Data Science, Springer, Cham, Switzerland, 2021.

8 M. Tsetlin, "On the behavior of finite automata in random media," Automation and Remote Control, pp. 1210–1219, 1961.

9 M. L. Tsetlin, "Automaton theory and the modeling of biological systems," New York: Academic, 1973.

10 K. S. Narendra and M. A. L. Thathachar, "Learning automata: A survey," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 4, pp. 323–334, 1974.

11 M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: An overview," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 32, pp. 711–722, 2002.

12 S. Lakshmivarahan, "Learning algorithms theory and applications," New York: Springer-Verlag, 1981.

13 K. S. Narendra and M. A. L. Thathachar, "Learning automata: An introduction," Englewood Cliffs, NJ: Prentice-Hall, 1989.

14 K. Najim and A. S. Poznyak, "Learning automata: Theory and applications," New York: Pergamon, 1994.

15 A. S. Poznyak and K. Najim, "Learning automata and stochastic optimization," New York: Springer, 1997.

16 Y. C. Ho, R. S. Sreenivas, and P. Vakili, "Ordinal optimization of discrete event dynamic systems," *Discrete Event Dynamic Systems (DEDS)*, vol. 2, no. 2, pp. 61–88, 1992.

17 H.-C. Chen, C.-H. Chen, and E. Yücesan, "Computing efforts allocation for ordinal optimization and discrete event simulation," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 960–964, 2000.

18 C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 10, pp. 251–270, 2000.

19 Y. Ho, Q. Zhao, and Q. Jia, "Ordinal optimization: Soft optimization for hard problems," New York: Springer, 2007.

20 J. Zhang, Z. Li, C. Wang, D. Zang, and M. Zhou, "Approximate simulation budget allocation for subset ranking," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 358–365, 2017.

21 J. Zhang, L. Zhang, C. Wang, and M. Zhou, "Approximately optimal computing budget allocation for selection of the best and worst designs," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3249–3261, 2017.

22 J. Zhang, C. Wang, D. Zang, and M. Zhou, "Incorporation of optimal computing budget allocation for ordinal optimization into learning automata," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1008–1017, 2016.